

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-279758

(43)公開日 平成8年(1996)10月22日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
H 0 3 M 7/30		9382-5K	H 0 3 M 7/30	B
G 1 0 L 7/04			G 1 0 L 7/04	G
9/18			9/18	H

審査請求 未請求 請求項の数 1 F D (全 14 頁)

(21)出願番号 特願平7-107957

(22)出願日 平成7年(1995)4月7日

(71)出願人 000000952

鐘紡株式会社

東京都墨田区墨田五丁目17番4号

(72)発明者 岡本 彰

大阪市都島区友測町1丁目6番7-304号

(72)発明者 下山 雅通

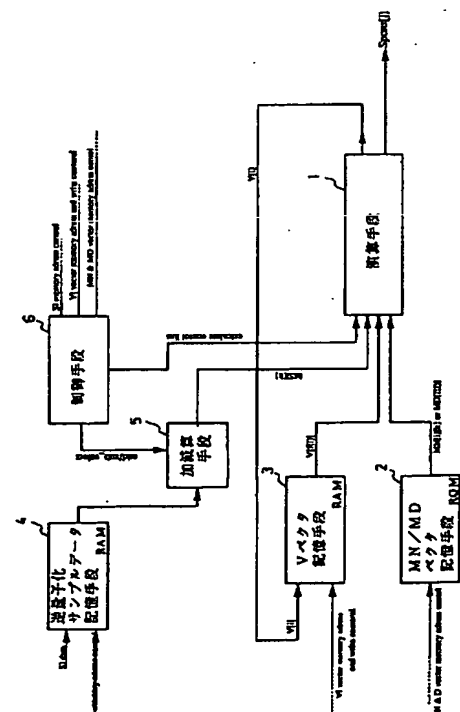
大阪市都島区友測町2丁目12番21-303号

(54)【発明の名称】 音声信号帯域合成復号化装置

(57)【要約】

【目的】MPEG規格を用いた圧縮音声／オーディオ信号の伸張において、ハードウェア規模の縮小、高速化を図ること。

【構成】サブバンド生成用のVベクタを算出する演算の冗長的な部分を削除して保持に必要な領域を縮小し、かつ係数の変換を行って、各演算手段に共通性を持たせ、同一のハードウェアで時分割処理を行う。



(2)

【特許請求の範囲】

【請求項 1】 MPEG規格を用いた圧縮音声信号の伸長装置であって、音声データPCMを算出する際に必要となる係数であるNベクタ及びDベクタに符号変換を行ったMDベクタを予め格納したN/MDベクタ記憶手段、逆量子化データを加算もしくは減算して修正逆量子化データを算出する加減算手段、修正逆量子化データ及びNベクタを用いたサブバンド合成用のVベクタの算出と、MDベクタ及びVベクタを用いた音声データPCMの算出とを同一演算法を用いて時分割処理により行う演算手段、Vベクタを記憶するVベクタ記憶手段と、これら各手段を制御する制御手段とからなることを特徴とする音声信号帯域合成複号化装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はMPEG規格の高効率符号化方式を用いた圧縮音声／オーディオの伸張装置に関し、更に詳しくは、比較的小規模なハードウェアで高速に伸張処理できる装置に関する。

【0002】

【従来の技術】最近では、マルチメディアへの使用を目的とした動画像／音声の高効率符号化が必要不可欠となっている。この高効率符号化の国際標準規格としてMPEG (Moving Picture Experts Group) 方式があり、既にISO/IEC 11172として勧告され、MPEG規格を用いた圧縮音声／オーディオ信号の伸張方法に関しては、「FAST Subband Filtering in MPEG Audio Coding (IEEE SIGNAL PROCESSING LETTER, Vol 1, NO 2, FEBRUARY 1994)」等にも記載されている。

【0003】かかるMPEG方式による圧縮音声／オーディオ伸張に際しての帯域合成処理の原理手順を説明する。図10はISO/IEC 11172の勧告に示されているMPEG方式のオーディオレイヤー2の規格にしたがった帯域合成のフローチャートである。MPEG方式では32のサンプルデータごとに逆量子化が行われ、帯域合成処理が開始される(ステップ101)。次のステップ103では生成される64個の新たなVベクタに備えて、1024個からなるVベクタ群中、古い64個がシフト処理され(ステップ102)、入力された32の逆量子化サンプルデータから新たなVベクタが64個算出される(ステップ103)。次に1024個のVベクトル群から512個のUベクタ群が生成され(ステップ104)、Uベクタ群にウインド処理に用いられる係数である係数D_iが乗じられて、ウインド処理されてWベクタ群が生成される(ステップ105)。Wベクタ群は、所定方向に16個加算されて1サンプル分の音声／オーディオデータ(32ワード)が伸張され(ス

2

テップ106)、これが36回繰り返して実行されて36グループからなる1フレームが伸張される。

【0004】また、特開平6-77839号公報等には、かかる演算やハードウェア規模を縮小するために、逆量子化ステップを量子化情報が共通な12グループ単位で演算を行い、1フレーム単位で一括して帯域合成処理を行う方法が提案されている。

【0005】かかる圧縮伸長を、前記した勧告の規格通りにハードウェアに適用した場合以下のような問題点が生ずる。音声データPCMサンプルの合成に際して、Vベクタ群、Uベクタ群、Wベクタ群を生成する必要があるため、処理手順が多く、使用するメモリも多くなる。各演算過程に共通性がないので回路の共有ができず、各演算に応じたハードウェアが必要となり、装置全体としてみればハードウェア規模が大きく、高価なものとなり民製品への利用の障害となる。

【0006】また、特開平6-77839号公報では、1フレーム単位で一括処理を行うことによってハードウェアの縮小を図っているが、逆量子化データ、Vベクタとして3264ワード分を必要とし、さらに1フレームごとの処理のため処理遅延が大きいという欠点が存在する。

【0007】本発明は前記課題を解決するため、サブバンド生成用のVベクタを算出する演算の冗長的な部分を削除して保持に必要な領域を縮小し、かつ係数(Dベクタ群)の変換を行って、各演算手順に共通性を持たせることによって、ハードウェア規模の縮小を図ることを目的とする。

【0008】

【課題を解決するための手段】本発明は、MPEG規格を用いた圧縮音声信号の伸長装置であって、音声データPCMを算出する際に必要となる係数であるNベクタ及びDベクタに符号変換を行ったMDベクタを予め格納したN/MDベクタ記憶手段、逆量子化データを加算もしくは減算して修正逆量子化データを算出する加減算手段、修正逆量子化データ及びNベクタを用いたサブバンド合成用のVベクタの算出と、MDベクタ及びVベクタを用いた音声データPCMの算出とを同一演算法を用いて時分割処理により行う演算手段、Vベクタを記憶するVベクタ記憶手段と、これら各手段を制御する制御手段とからなることを特徴とする音声信号帯域合成複号化装置である。

【0009】

【作用】本装置では、逆量子化データを修正して用いると共に音声データPCMを算出する際に必要とされるDベクタを変換して用いることにより、サブバンド合成用のVベクタの算出手法と音声データPCMの演算法を同一のものとし、両手段を共用する。

【0010】即ち、Vベクタの算出は後述するNベクタの変換と逆量子化データSの修正を行うことにより次式

(3)

で求められる。

【0011】

【数1】

$0 \leq i \leq 31$

$$V512[i] = \sum_{k=0}^{15} (MN[i][K] * MSi(K))$$

$0 \leq i \leq 31$

$$Sj[j] = \sum_{i=0}^{15} [V512(F1(i, j) * MD(F2(i, j)))]$$

【0014】両式は、乗算とシグマ加算とからなるものであるため、同じ演算手法（演算回路）を共用することができる。

【0015】

【実施例】以下、本発明の帯域合成複合化装置について詳述する。図1は本装置の基本的な装置構成を示すブロック図である。同図に示す如く、本装置は、サブバンド合成用のVベクタの算出手段と音声データPCMの演算手段を共用する演算手段1を中心として、該演算手段にデータを入力する手段として、MN/MDベクタ記憶手段2、演算手段から出力されるVベクタを格納するVベクタ記憶手段3、逆量子化サンプルデータを格納した逆量子化サンプルデータメモリ4からの出力を加減算して修正逆量子化サンプルデータを出力する加減算手段5を有し、これら各手段のコントロールを行う制御手段6とからなる。

【0016】以下、各手段について、順次詳述する。図2は、逆量子化サンプルデータメモリ4の内部構成を示すブロック図である。このブロックは逆量子化されたサンプルデータを保持するブロックで、1グループ分（36グループで1フレームを構成）を保持する32ワードのメモリ41（RAM）と、メモリのリード/ライトを制御するコントローラ42が存在し、出力するデータのアドレスは制御手段により制御される。

【0017】加減算手段5は、Vベクタを算出する前処理として、逆量子化されたデータを加算/減算するものであり、図3は、同手段の内部構成を示すブロック図である。逆量子化サンプルデータは、後述する式に基づいて特定アドレスの逆量子化サンプルデータと加算又は減算が施されるが、加減算手段には、1ワードづつ入力を行うのが装置規模を縮小するためには好ましく、このため加算/減算器51の一方にはレジスタ52が設けられ、制御装置からの指示によって、連続して入力された2つの逆量子化サンプルデータが加算/減算処理される。

【0018】Vベクタ記憶手段3は、後述する演算装置によって算出されたVベクタを保持するメモリ31で、512ワードのメモリに、新規作成される32個のVベクタを含む512個のVベクタを保持する。Vベクタ記

4

【0012】一方、Dベクタを変換したMDベクタを用いると、音声データPCM（Sj）の算出は次式により求めることができる。

【0013】

【数2】

憶手段の内部構成を図4に示す。同手段は、メモリを初期設定（0番地から480番地までデータ値をゼロに設定）するブロック32と、Vベクタのアドレス番地を書き換える（シフティング）ブロック33、メモリのライト/リード動作を制御するメモリコントロールブロック34で構成されており、メモリに使用するアドレス値や初期設定命令、シフティング命令は制御手段から制御される。

【0019】MN/MDベクタ記憶手段2は、乗算に使用する係数であり勧告で定められたNベクタ及びDベクタについて、これを変換したMNベクタ及びMDベクタを予め格納するもので、本実施例では、ROMを用いている。MN/MDベクタ記憶手段の内部構成を図5に示す。

【0020】MNベクタは後述するNベクタの冗長的な係数部分を削除したものであり、MDベクタはDベクタに、後述する符号変換関数を用いて符号変換を行ったものであり、さらに、Vベクタと同様のアドレス関数を使用することができるようにアドレスの番地の変換も施したものであり、該アドレスは制御装置により制御される。

【0021】制御手段6は、本装置の各種制御である初期設定やシフティング、積和演算、メモリのアドレスや動作制御等を統括する手段であり、その内部構成を図6に示す。同図に示す如く、本実施例では、外部から帯域合成スタート命令を受けて、マスタカウンタ61を稼働させ、カウンタ値をデコードして、逆量子化サンプルデータメモリ制御デコーダ62、加減算手段制御デコーダ63、Vベクタ記憶手段制御デコーダ64、N/MDベクタ記憶手段制御デコーダ65、演算手段制御デコーダ66の夫々を起動する信号を発生する。

【0022】各デコーダはまずVベクタ記憶手段デコーダを起動し、後述する図11ステップ201のシフティング動作を行う。次に、逆量子化サンプルメモリ制御デコーダ62、加減算手段制御デコーダ63、Vベクタ記憶手段デコーダ64、MN/MDベクタ65の記憶手段デコーダ及び演算手段制御デコーダ66を起動し、図11ステップ202のVベクタ算出/記憶を行う。次にVベクタ記憶手段デコーダ64、MN/MDベクタの記憶

(4)

5

手段デコーダ65及び演算手段制御デコーダ66を起動し、図11ステップ203の出力PCM信号を算出する。

【0023】演算手段1は、Vベクタの算出及びPCMデータの算出に使用する積和演算器であり、その内部構成を図7に示す。本発明では、後述する係数等を採用することにより2つの積和演算は同一の演算手法、演算量となるため、演算手段では、時分割処理を行い同一回路を用いて演算を行う。よって入力値を選択するセレクタ12a、12bを設けて修正逆量子化サンプルデータとMNベクタの組み合わせと、VベクタとMDベクタの組み合わせを選択する。また、積和演算器11の制御やセレクタライン等は制御手段により制御されている。

【0024】次に、本装置で用いる帯域合成のアルゴリズムについて説明する。かかるアルゴリズムは、ISO/IEC 11172の勧告に示されているMPEG方式のオーディオレイヤー2の規格にしたがった帯域合成の演算手法を基本として、次の4項目について改良を加えることを特徴としている。以下、各改良項目について詳述する。

【0025】(1) Vベクタの算出

図10の勧告に示される処理フローでは、まず前回までのVベクタ群のアドレス位置を変更する処理(shifting)が行われた後、新規のVベクタ(64個)が算出される。勧告に示されるVベクタの算出は以下の式の如くである。

【0026】

【数3】

$$0 \leq i \leq 15, 33 \leq i \leq 47$$

$$k = 2n \quad \text{のとき} +$$

$$k = 2n + 1 \quad \text{のとき} -$$

$$V[i] = \sum_{k=0}^{15} (N[i][k] * (S[k] \pm S[31-k]))$$

$$i = 16$$

$$V[16] = 0$$

$$17 \leq i \leq 32$$

$$V[i] = -V[32-i]$$

$$i = 48$$

$$V[i] = \sum_{k=0}^{15} (S[k] + S[31-k])$$

$$49 \leq i \leq 63$$

$$V[i] = V[96-i]$$

【0032】結果、必要とする演算量は、496回(16×31)の乗算と480回(15×32)の和、及び前処理としてそれぞれ16回の和・差、及び後処理としての符号変換が16回となり、演算に必要なNベクタは図8に示された32行×16列のマトリクスとなり、こ

6

$$0 \leq i \leq 63$$

$$V[i] = \sum_{k=0}^{31} (N[i][k] * S[k])$$

【0027】また、この式に用いられる係数N[i][k]は以下の式によって定義されている。

【0028】

【数4】

$$0 \leq i \leq 63 \quad 0 \leq k \leq 31$$

$$N[i][k] = \cos[(16+i)(2k+1)\pi/64]$$

【0029】かかる式から分かるように新規のVベクタを算出するには、2048回(32×64)の乗算と1984回(31×64)の和が必要となってくる。ところが、この係数N[i][k]を[i=行][k=列]としたマトリクス形式で表現すると以下の様な性質があることが解る。

【0030】17行から32行までの係数は、0行から15行までの係数と対象であり、逆符号の関係である。

49行から63行までの係数は、33行から47行までの係数と対象の関係である。16行目の係数は常に「零(0)」であり、48行目の係数は「-1」である。16列から31列までの係数は、0列から15列までの係数と奇数行が対象、偶数行が逆符号対象の関係である。以上の性質を利用し、Vベクタの算出は図8に示すように変形され、次式の様に変形される。

【0031】

【数5】

れをMNベクタと定義する。

【0033】(2) Vベクタ群の履歴の保持

Vベクタは、勧告では新規に作成された64個のVベクタと過去15回分のVベクタ(64×15=980)の合わせて1024個のVベクタを保持する必要がある。

(5)

しかし、前述の如く⁷Vベクタは、符号だけが違い、絶対値は同じであるデータが約半数を占めているため、新規に作成されたVベクタを例にとると、図8に示されるようにデータ内容の重複あるいは符号が反転しているものが存在することが確認でき、保持すべきVベクタは64個の内、半分の32個で良いことが解る。よって全体では保持すべきVベクタ群は1024個(64×16)の内、512個(32×16)となり、シフトの回数も9*

0 ≤ k ≤ 63 とすると

$$\begin{aligned}
 V1024[64*N+k] &= V512[32*N+k] & (0 \leq k \leq 15) \\
 &= 0 & (k=16) \\
 &= V512[32*N+(32-k)] & (17 \leq k \leq 32) \\
 &= V512[32*N+(k-17)] & (33 \leq k \leq 48) \\
 &= V512[32*N+(79-k)] & (49 \leq k \leq 63)
 \end{aligned}$$

0 ≤ k ≤ 31 とすると

$$\begin{aligned}
 V1024[64*N+k] &= V512[32*N+k] & (0 \leq k \leq 15) \\
 &= 0 & (k=16) \\
 &= V512[32*N+(32-k)] & (17 \leq k \leq 32) \\
 V1024[64*N+32+k] &= V512[32*N+k] & (k=0) \\
 &= V512[32*N+(k+15)] & (1 \leq k \leq 16) \\
 &= V512[32*N+(47-k)] & (17 \leq k \leq 31)
 \end{aligned}$$

【0035】(3) 中間ベクタ(Uベクタ・Wベクタ)作成の削除
 勧告によるフローでは、作成されたVベクタから中間的なベクタ群(Uベクタ・Wベクタ)を抽出・算出して音声データPCMを算出する。勧告による音声データPCMを算出する式を以下に示す。

【0036】

【数7】

$$0 \leq j \leq 31$$

$$S_j[j] = \sum_{i=0}^{15} W(j+32*i)$$

$$0 \leq i \leq 7$$

$$0 \leq j \leq 31$$

$$U[i*64+j] = V1024[i*128+j]$$

$$U[i*64+32+j] = V1024[i*128+96+j]$$

$$0 \leq i \leq 511$$

$$W[i] = U[i] * [i]$$

【0039】この結果、前記式は以下の様に変換される。

⁸*60回から480回となり、処理時間、保持領域メモリを半減せしめることが可能となる。前述の性質より、1024個保持するVベクタ群をV1024、512個保持するVベクタ群をV512とすると、2つのベクタ群の関係は次の式の如くとなる。

【0034】

【数6】

【0037】この式に対して、以下の式を用いて変形し、それぞれのベクタのアドレスを示す関数(F1、F2、F3)を定義する。

【0038】

【数8】

【0040】

【数9】

(6)

9
0 <= j <= 31

10

$$S_j[j] = \sum_{i=0}^{15} [F_3(i, j) * V_{512}(F_1(i, j)) * D(F_2(i, j))]$$

F1, F2: アドレス関数

F3 : 符号変換関数

【0041】また、この変形過程を図9に示す。かかる式に定義したアドレス関数を用いればV512ベクタ群からU/Wベクタに相当するデータを読み出すことができるので、U/Wといった中間ベクタ群を生成する手順を省略できる。

【0042】(4) Dベクタ群の改良

前記式を用いて音声データPCMを算出する場合、V512ベクタから抽出したベクタに符号変換関数(F3関数)を用いて符号変換を行った後、Dベクタと乗算する。符号変換関数およびDベクタは、V512ベクタと*

0 <= j <= 31

$$S_j[j] = \sum_{i=0}^{15} [V_{512}(F_1(i, j)) * MD(F_2(i, j))]$$

【0044】図11は、以上の改良を採り入れた本装置で用いる帯域合成のアルゴリズムを示す処理フローチャートである。先ず、逆量子化されたデータが1グループ(32ワード)分入力される(ステップ201)。而して、以下のステップでは、前記(1)、(2)の項で説明したデータの性質を用いて、特開平6-77839号公報がフレーム単位で処理を行っていたのに対し、32個のデータを単位として処理を行う。即ち、ステップ202では生成される32個の新たなVベクタに備えて、512個からなるVベクタ群中、古い32個がシフト処理され、入力された逆量子化サンプルデータから新たなVベクタが32個算出される(ステップ203)。従って、新たに保持すべきVベクタは512個で良い。

【0045】次に前記(3)(4)の項で説明したアルゴリズムを用いて、Uベクタ群、Wベクタ群の生成を省略して1グループ分の音声/オーディオデータ(32PCMサンプルデータ)が伸張され(ステップ203)、これが36回繰り返して実行されて1フレームが伸張される。

【0046】以上の如きアルゴリズムでは、MPEG方式規格を準拠しつつ冗長的な演算を削除したため、ハードウェア化した場合、以下のような特徴を有する。係数Nの冗長性を利用した積和演算を行うために演算量が半減し高速処理が可能となる。保持するデータ(Vベクタ)の冗長的な部分を保持しておく必要がないため、保持領域が半減し、メモリアクセス量を減少させることができる。音声データを生成する際に必要となる中間ベクタ(U/Wベクタ)を生成する必要がないため処理フローを簡素化することできる。Vベクタの算出方式にお

* 同一アドレスで一対一に対応する性質を有するので、符号変換関数とDベクタを乗算したものを保持するだけで、Vベクタの符号変換演算を削除できる。よって符号変換関数とDベクタを乗算した結果を、新係数MDベクタ(modified Dvector)と定義すると、以下の式の様になり、Vベクタを算出したときと同様の積和演算で済むこととなる。

【0043】

【数10】

る冗長部分の削除およびDベクタを符号変換した新係数の採用により、音声データを作成する演算がVベクタを作成する過程と同様の積和演算(乗算とシグマ加算)となるため、演算器を共有することが可能になり、ハードウェアの構成を簡素化することができる。

【0047】

【発明の効果】本発明では、帯域合成復号化装置の冗長的な演算処理を削除したアルゴリズムを採用することにより、デコード期間の短縮/高速化と、ハードウェア規模の縮小化が図れ、同等の機能を安価に製造することができ、本装置は、MPEG規格による圧縮音声の伸張を民生用途に利用する際に頗る有用なものである。

【図面の簡単な説明】

【図1】本装置全体の基本的な構成を示すブロック図である。

【図2】本装置の逆量子化サンプルデータメモリの内部構成を示すブロック図である

【図3】本装置の加減算手段の内部構成を示すブロック図である。

【図4】本装置のVベクタ記憶手段の内部構成を示すブロック図である。

【図5】本装置のN/MDベクタ記憶手段の内部構成を示すブロック図である。

【図6】本装置の制御手段の内部構成を示すブロック図である。

【図7】本装置の演算手段の内部構成を示すブロック図である。

【図8】Vベクタの算出過程を示す説明図である。

【図9】Vベクタの算出過程を示す説明図である。

(7)

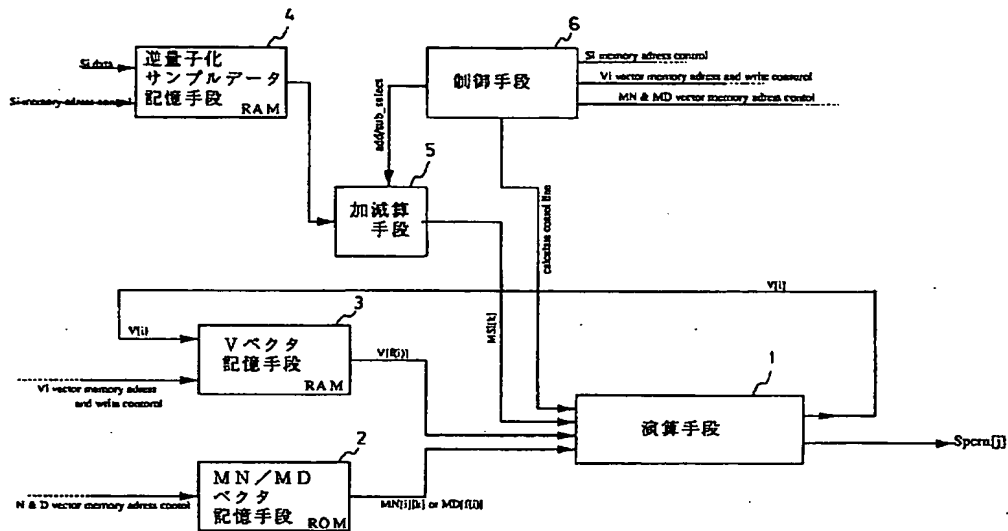
11

【図10】MPEG方式の勧告に示されている帯域合成処理手順を示すフローチャートである。

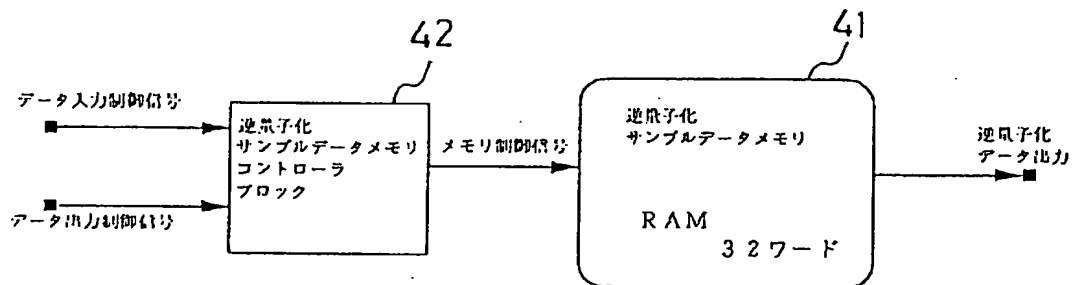
12

【図11】本装置で用いる帯域合成処理手順を示すフローチャートである。

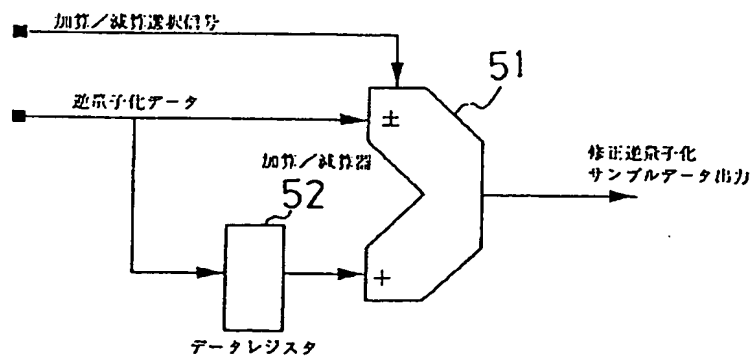
【図1】



【図2】

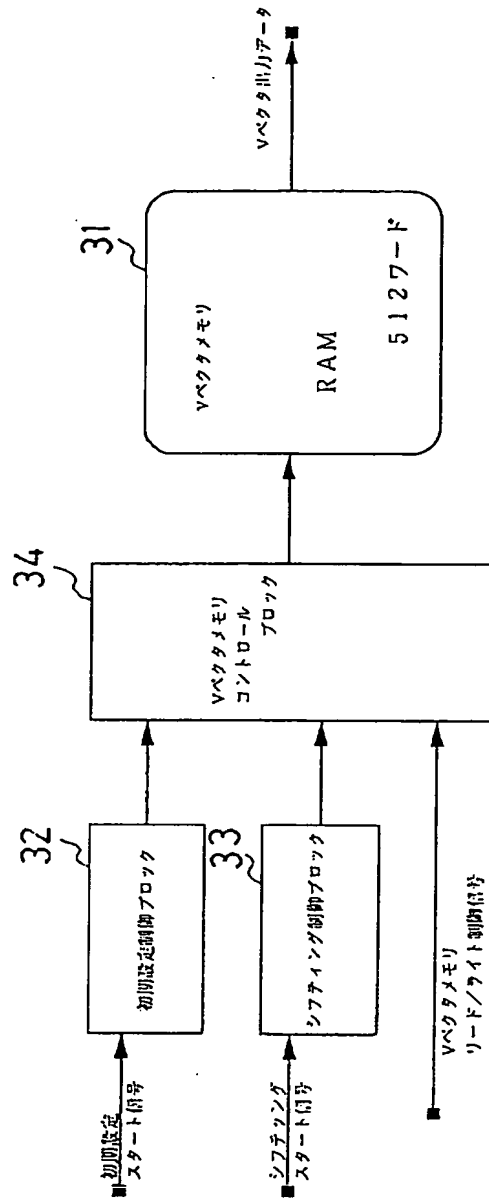


【図3】



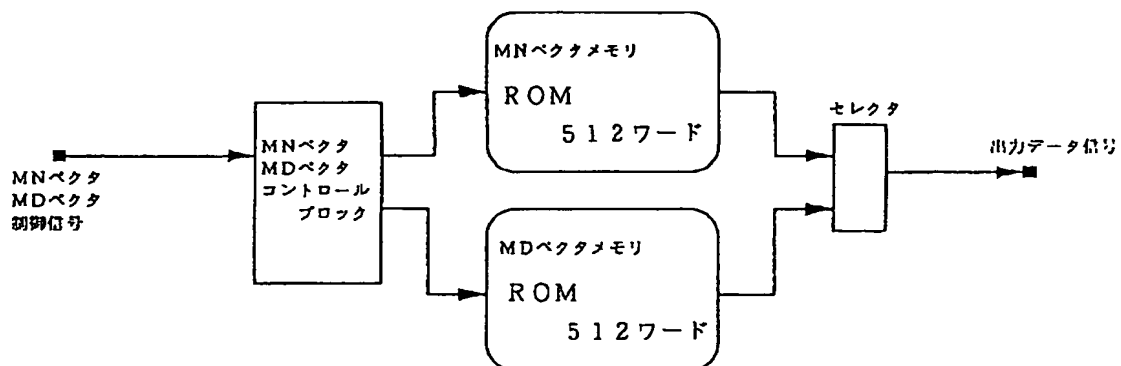
(8)

【図4】

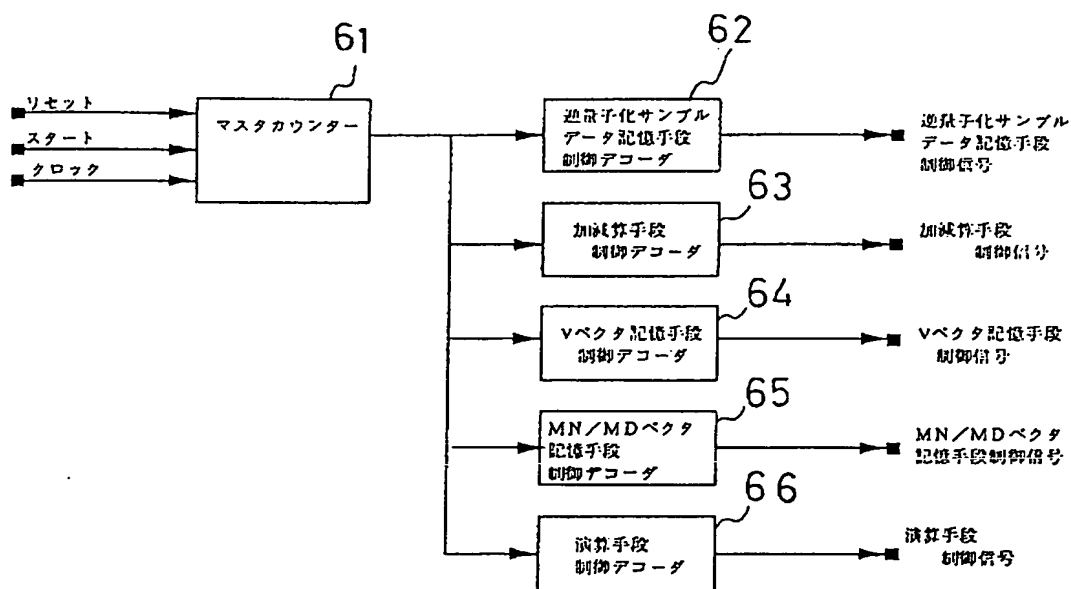


(9)

【図5】

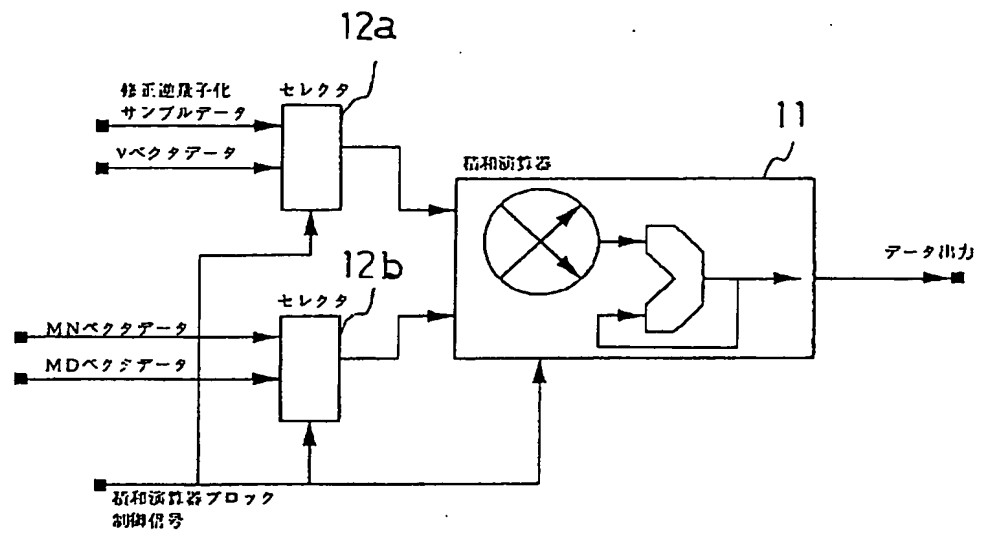


【図6】



(10)

【図7】



(11)

【図8】

係数 $N[i][k]$ の性質 (4) を利用して以下のように変形する

$$\begin{pmatrix} V[0] \\ V[1] \\ \vdots \\ V[2n] \\ V[2n+1] \\ \vdots \\ V[63] \end{pmatrix} = \begin{pmatrix} N[0,0] & N[0,1] & \cdots & N[0,14] & N[0,15] & N[0,15] & N[0,14] & \cdots & N[0,1] & N[0,0] \\ N[1,0] & N[1,1] & \cdots & N[1,14] & N[1,15] & -N[1,15] & -N[1,14] & \cdots & -N[1,1] & -N[1,0] \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ N[2n,0] & \cdots & N[2n,15] & N[2n,15] & \cdots & N[2n,0] \\ N[2n+1,0] & \cdots & N[2n+1,15] & -N[2n+1,15] & \cdots & -N[2n+1,0] \\ \vdots & & \vdots & \vdots & & \vdots \\ N[63,0] & N[63,1] & \cdots & N[63,14] & N[63,15] & -N[63,15] & -N[63,14] & \cdots & -N[63,1] & -N[63,0] \end{pmatrix} \begin{pmatrix} S[0] \\ S[1] \\ \vdots \\ S[2n] \\ S[2n+1] \\ \vdots \\ S[31] \end{pmatrix}$$

$$\begin{pmatrix} V[0] \\ V[1] \\ \vdots \\ V[63] \end{pmatrix} = \begin{pmatrix} N[0,0] & N[0,1] & \cdots & N[0,14] & N[0,15] \\ N[1,0] & & & & \vdots \\ \vdots & & & & \vdots \\ N[63,0] & \cdots & & & N[63,15] \end{pmatrix} \begin{pmatrix} S[0] \pm S[31] \\ S[1] \pm S[30] \\ \vdots \\ S[n] \pm S[31-n] \\ \vdots \\ S[15] \pm S[16] \end{pmatrix}$$

偶数行は「+」
奇数行は「-」をとる

次に係数 $N[i][k]$ の性質 (1)(2)(3) を利用して以下のように変形する

$$\begin{pmatrix} V[0] \\ \vdots \\ V[15] \\ V[16] \\ V[17] \\ \vdots \\ V[32] \\ V[33] \\ \vdots \\ V[47] \\ V[48] \\ V[49] \\ \vdots \\ V[63] \end{pmatrix} = \begin{pmatrix} N[0,0] & N[0,1] & \cdots & N[0,14] & N[0,15] \\ \vdots & \vdots & & \vdots & \vdots \\ N[15,0] & \cdots & & & \\ 0 & \cdots & & & \\ -N[15,0] & \cdots & & & \\ \vdots & \vdots & & \vdots & \vdots \\ -N[0,0] & \cdots & & & \\ N[33,0] & \cdots & & & \\ \vdots & \vdots & & \vdots & \vdots \\ N[47,0] & \cdots & & & \\ -1 & \cdots & & & \\ N[47,0] & \cdots & & & \\ \vdots & \vdots & & \vdots & \vdots \\ N[33,0] & \cdots & & & N[33,15] \end{pmatrix} \begin{pmatrix} S[0] \pm S[31] \\ S[1] \pm S[30] \\ \vdots \\ S[n] \pm S[31-n] \\ \vdots \\ S[15] \pm S[16] \end{pmatrix}$$

$$\begin{pmatrix} V[0] \\ \vdots \\ V[15] \\ V[33] \\ \vdots \\ V[47] \\ V[48] \end{pmatrix} = \begin{pmatrix} N[0,0] & N[0,1] & \cdots & N[0,14] & N[0,15] \\ \vdots & \vdots & & \vdots & \vdots \\ N[15,0] & \cdots & & & \\ N[33,0] & \cdots & & & \\ \vdots & \vdots & & \vdots & \vdots \\ N[47,0] & \cdots & & & N[47,15] \\ -1 & \cdots & & & -1 \end{pmatrix} \begin{pmatrix} S[0] \pm S[31] \\ S[1] \pm S[30] \\ \vdots \\ S[15] \pm S[16] \end{pmatrix}$$

$i = 16$

$V[i] = 0$

$17 \leq i \leq 32$

$V[i] = -V[32-i]$

$49 \leq i \leq 63$

$V[i] = V[96-i]$

偶数行は「+」

奇数行は「-」をとる

(12)

【図9】

$$\begin{aligned}
 j = 0 \text{ to } 31 \quad S_j &= \sum_{i=0}^{15} W(j+32i) \\
 &= W(j+32*0) + W(j+32*1) + W(j+32*2) + \dots + W(j+32*j) + \dots + W(j+32*15) \\
 &= W(j+64*0) + W(j+64*0+32) + W(j+64*1) + W(j+64*1+32) + \dots + W(j+64*7) + W(j+64*7+32) \\
 &= \sum_{i=0}^7 [W(j+64*i) + W(j+64*i+32)]
 \end{aligned}$$

Wベクタを差し替える

$$= \sum_{i=0}^7 [U(64*i+j) * D(64*i+j) + U(64*i+32+j) * D(64*i+32+j)]$$

さらにUベクタをVs12ベクタに差し替える

if (j=0) then

$$= \sum_{i=0}^7 [Vs12(32*2i+j) * D(j+64*i) + Vs12(32*(2i+1)) * D(j+64*i+32)]$$

if (1 <= j <= 15) then

$$= \sum_{i=0}^7 [Vs12(32*2i+j) * D(j+64*i) + Vs12(32*(2i+1)+j+15) * D(j+64*i+32)]$$

if (j=16) then

$$= \sum_{i=0}^7 [0 * D(j+64*i) + Vs12(32*(2i+1)+j+15) * D(j+64*i+32)]$$

if (17 <= j <= 31) then

$$= \sum_{i=0}^7 [-Vs12(32*2i+(32-j)) * D(j+64*i) + Vs12(32*(2i+1)+(47-j)) * D(j+64*i+32)]$$

各アドレスを示す関数 (変数 i, j) を定義する

(1) F1(i,j)関数の定義

i = 2n and 0 <= j <= 16

$$F1(i,j) = 32*i + j$$

i = 2n and 17 <= j <= 31

$$F1(i,j) = 32*i + 32 - j$$

i = 2n+1 and j = 0

$$F1(i,j) = 32*i$$

i = 2n+1 and 1 <= j <= 16

$$F1(i,j) = 32*i + j + 15$$

i = 2n+1 and 17 <= j <= 31

$$F1(i,j) = 32*i + 47 - j$$

(2) F2(i,j)関数の定義

i = 2n

$$F1(i,j) = 64*i + j$$

i = 2n+1

$$F1(i,j) = 32*i + j + 32$$

(3) F3(i,j)関数の定義

i = 2n and 0 <= j <= 15

$$F3(i,j) = 1$$

i = 2n and j = 16

$$F3(i,j) = 0$$

i = 2n and 17 <= j <= 31

$$F3(i,j) = -1$$

i = 2n+1 and j = 0

$$F3(i,j) = -1$$

i = 2n+1 and 1 <= j <= 31

$$F3(i,j) = 1$$

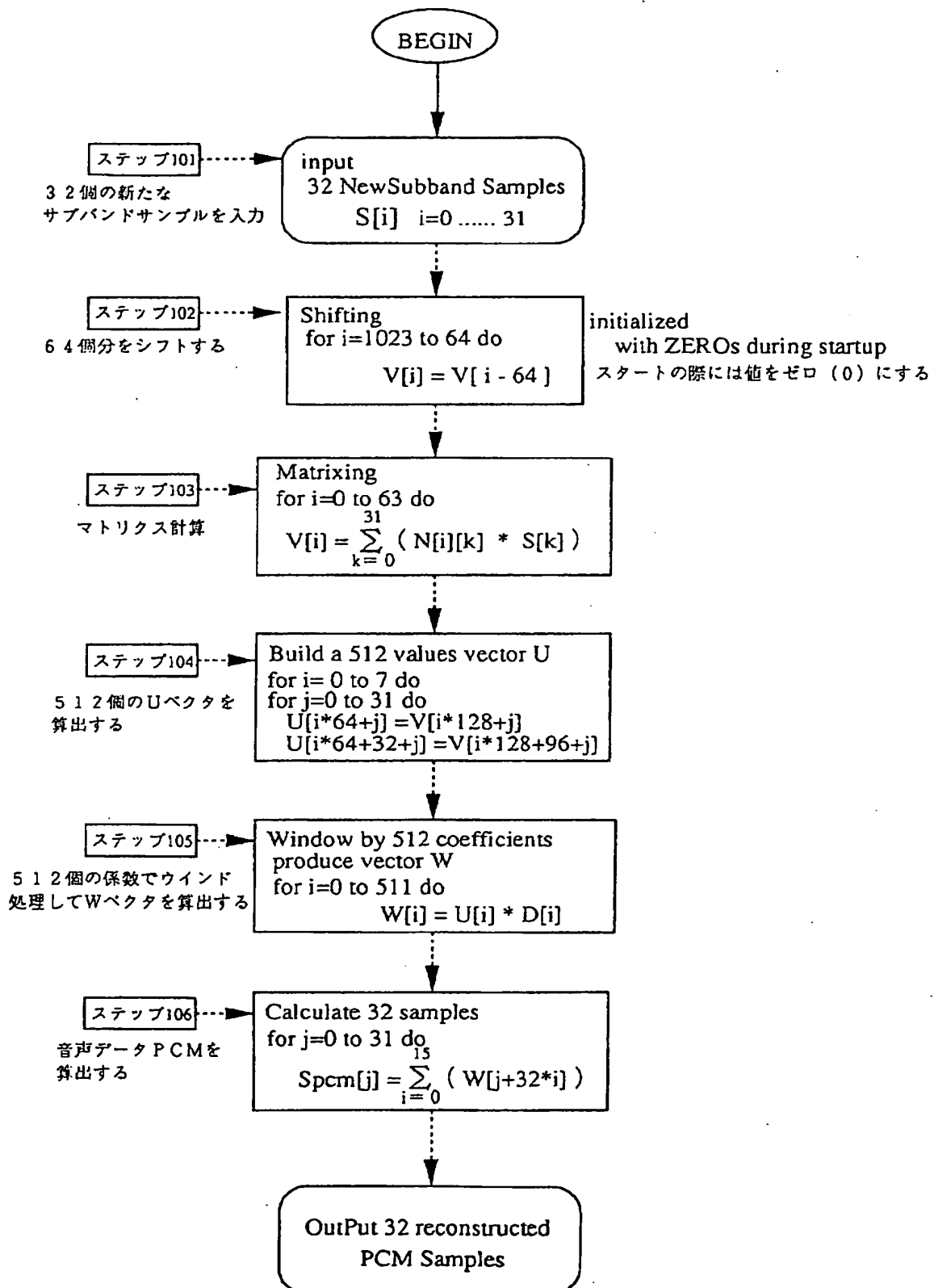
結果、算出式は以下ようになる

j = 0 to 31

$$S(j) = \sum_{i=0}^{15} [F3(i,j) * Vs12(F1(i,j)) * D(F2(i,j))]$$

(13)

【図10】



(14)

【図11】

